A Framework for Comparing Groups of Documents

Arun S. Maiya
Institute for Defense Analyses — Alexandria, VA, USA
amaiya@ida.org

Abstract

We present a general framework for comparing multiple groups of documents. A bipartite graph model is proposed where document groups are represented as one node set and the comparison criteria are represented as the other node set. Using this model, we present basic algorithms to extract insights into similarities and differences among the document groups. Finally, we demonstrate the versatility of our framework through an analysis of NSF funding programs for basic research.

1 Introduction and Motivation

Given multiple sets (or groups) of documents, it is often necessary to *compare* the groups to identify similarities and differences along different dimensions. In this work, we present a general framework to perform such comparisons for extraction of important insights. Indeed, many real-world tasks can be framed as a problem of comparing two or more *groups* of documents. Here, we provide two motivating examples.

1. Program Reviews. To better direct research efforts, funding organizations such as the National Science Foundation (NSF), the National Institutes of Health (NIH), and the Department of Defense (DoD), are often in the position of reviewing research programs via their artifacts (*e.g.*, grant abstracts, published papers, and other research descriptions). Such reviews might involve identifying overlaps across different programs, which may indicate a duplication of effort. It may also involve the identification of unique, emerging, or diminishing topics. A "document group" here could be defined either as a particular research program that funds many organizations, the totality of funded research conducted by a specific organization, or all research associated with a particular time period (*e.g.*, fiscal year). In all cases, the objective is to draw comparisons *between* groups by comparing the document sets associated with them.

2. Intelligence. In the areas of defense and intelligence, document sets are sometimes obtained from different sources or entities. For instance, the U.S. Armed Forces sometimes seize documents during raids of terrorist strongholds.[1] Similarities between two document sets (each captured from a different source) can potentially be used to infer a non-obvious association between the sources.

Of course, there are numerous additional examples across many domains (*e.g.,* comparing different news sources, comparing the reviews for several products, etc.). Given the abundance of real-world applications as illustrated above, it is surprising, then, that there are no existing general-purpose approaches for drawing such comparisons. While there is some previous work on the comparison of document sets (referred to as *comparative text mining*), these existing approaches lack the generality to be widely applicable across different use case scenarios with different comparison criteria. Moreover, much of the work in the area focuses largely on the summarization of shared or unshared topics among document groups (*e.g.,* Wan et al. (2011), Huang et al. (2011), Campr and Ježek (2013), Wang et al. (2012), Zhai et al. (2004)). That is, the problem of drawing *multi-faceted* comparisons among the groups themselves is not typically addressed. This, then, motivates our development of a *general-purpose* model for comparisons of document sets along arbitrary dimensions. We use this model for the identification of similarities, differences, trends, and anomalies among large *groups* of documents. We begin by formally describing our model.

2 Our Formal Model for Comparing Document Groups

As input, we are given several groups of documents, and our task is to compare them. We now formally define these document groups and the criteria used to compare them. Let $D = \{d_1, d_2, \ldots, d_N\}$ be a document collection comprising the totality of documents under consideration, where N is the size. Let D^P be a partition of D representing the document groups.

[1] See *Document Exploitation (DOCEX)* at http://en.wikipedia.org for more information.

Definition 1 *A document group is a subset $D_i^P \in D^P$ (where index $i \in \{1 \ldots |D^P|\}$).*

Each document group in D^P, for instance, might represent articles associated with either a particular organization (*e.g.,* university), a research funding source (*e.g.,* NSF or DARPA program), or a time period (*e.g.,* a fiscal year). Document groups are compared using *comparison criteria*, D^C, a family of subsets of D.

Definition 2 *A comparison criterion is a subset $D_i^C \in D^C$ (where index $i \in \{1 \ldots |D^C|\}$).*

Intuitively, each subset of D^C represents a set of documents sharing some attribute. Our model allows great flexibility in how D^C is defined. For instance, D^C might be defined by the named entities mentioned within documents (*e.g.,* each subset contains documents that mention a particular person or organization of interest). For the present work, we define D^C by topics discovered using latent Dirichlet allocation or LDA (Blei et al., 2003).

LDA Topics as Comparison Criteria. Probabilistic topic modeling algorithms like LDA discover latent themes (*i.e.,* topics) in document collections. By using these discovered topics as the comparison criteria, we can compare arbitrary groups of documents by the themes and subject areas comprising them. Let K be the number of topics or themes in D. Each document in D is composed of a sequence of words: $d_i = \langle s_{i1}, s_{i2}, \ldots, s_{iN_i} \rangle$, where N_i is the number of words in d_i and $i \in \{1 \ldots N\}$. $V = \bigcup_{i=1}^{N} f(d_i)$ is the vocabulary of D, where $f(\cdot)$ takes a sequence of elements and returns a set. LDA takes K and D (including its components such as V) as input and produces two matrices as output, one of which is θ. The matrix $\theta \in \mathbb{R}^{N \times K}$ is the document-topic distribution matrix and shows the distribution of topics within each document. Each row of the matrix represents a probability distribution. D^C is constructed using K subsets of documents, each of which represent a set of documents pertaining largely to the same topic. That is, for $t \in \{1 \ldots K\}$ and $i \in \{1 \ldots N\}$, each subset $D_t^C \in D^C$ is comprised of all documents d_i where $t = \text{argmax}_x \, \theta_{ix}$.[2] Having defined the document groups D^P and the comparison criteria D^C, we now construct a bipartite graph model used to perform comparisons.

A Bipartite Graph Model. Our objective is to compare the *document groups* in D^P based on D^C. We do so by representing D^P and D^C as a weighted bipartite graph, $G = (P, C, E, w)$, where P and C are disjoint sets of nodes, E is the edge set, and $w : E \rightarrow \mathbb{Z}^+$ are the edge weights. Each subset of D^P is represented as a node in P, and each subset of D^C is represented as a node in C. Let $\alpha : P \rightarrow D^P$ and $\beta : C \rightarrow D^C$ be functions that map nodes to the document subsets that they represent. Then, the edge set E is $\{(u, v) \mid u \in P, v \in C, \alpha(u) \cap \beta(v) \neq \emptyset\}$, and the edge weight for any two nodes $u \in P$ and $v \in C$ is $w((u, v)) = |\alpha(u) \cap \beta(v)|$. Concisely, each weighted edge in G between a document group (in P) and a topic (in C) represents the number of documents shared among the two sets. Figure 1 shows a toy illustration of the model. Each node in P is shown in black and represents a subset of D^P (*i.e.,* a document group). Each node in C is shown in gray and represents a subset of D^C (*i.e.,* a document cluster pertaining primarily to the same topic). Each edge represents the intersection of the two subsets it connects. In the next section, we will describe basic algorithms on such bipartite graphs capable of yielding important insights into the similarities and differences among document groups.

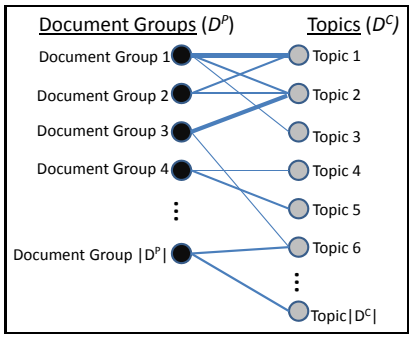

Figure 1: **[Toy Illustration of Bipartite Graph Model.]** Each black node (*i.e.,* node $\in P$) represents a document group. Each gray node (*i.e.,* node $\in C$) represents a cluster of documents pertaining primarily to the same topic.

3 Basic Algorithms Using the Model

We focus on three basic operations in this work.

Node Entropy. Let $\vec{w}$ be a vector of weights for all edges incident to some node $v \in E$. The *entropy* H of v is: $\text{H}(v) = -\sum_i p_i \log_{|\vec{w}|}(p_i)$, where $p_i = \frac{w_i}{\sum_j w_j}$ and $i, j \in \{1 \ldots |\vec{w}|\}$. A similar formulation was employed in Eagle et al. (2010). Intuitively, if $v \in P$, $H(v)$ measures the extent to which the document group is concentrated around a small number of topics (lower values of $H(v)$ mean more concentrated). Similarly, if $v \in C$, it is the extent to which a topic is concentrated around a small number of document groups.

Node Similarity. Given a graph G, there are many ways to measure the similarity of two nodes based on their connections. Such measures can be used to infer similarity (and dissimilarity) among document groups. However, existing methods are not well-suited for the task of document group comparison. The well-

[2] D^C is also a partition of D, when defined in this way.

known SimRank algorithm (Jeh and Widom, 2002) ignores edge weights, and neither SimRank nor its extension, SimRank++ (Antonellis et al., 2008), scale to larger graphs. SimRank++ and ASCOS (Chen and Giles, 2013) do incorporate edge weights but in ways that are not appropriate for document group comparisons. For instance, both SimRank++ and ASCOS incorporate magnitude in the similarity computation. Consider the case where document groups are defined as research labs. ASCOS and SimRank++ will measure large research labs and small research labs as less similar when in fact they may publish nearly identical lines of research. Finally, under these existing methods, document groups sharing zero topics in common could still be considered similar, which is undesirable here. For these reasons, we formulate similarity as follows. Let $N^G(\cdot)$ be a function that returns the neighbors of a given node in G. Given two nodes $u, v \in P$, let $L^{u,v} = N^G(u) \cup N^G(v)$ and let $x : I \rightarrow L^{u,v}$ be the indexing function for $L^{u,v}$.[3] We construct two vectors, $\vec{a}$ and $\vec{b}$, where $a_k = w(u, x(k))$, $b_k = w(v, x(k))$, and $k \in I$. Each vector is essentially a sequence of weights for edges between $u, v \in P$ and each node in $L^{u,v}$. Similarity of two nodes is measured using the cosine similarity of their corresponding sequences, $\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$, which we compute using a function $sim(\cdot, \cdot)$. Thus, document groups are considered more similar when they have similar sets of topics in similar proportions. As we will show later, this simple solution, based on item-based collaborative filtering (Sarwar et al., 2001), is surprisingly effective at inferring similarity among document groups in G.

Node Clusters. Identifying clusters of related nodes in the bipartite graph G can show how document groups form larger classes. However, we find that G is typically fairly dense. For these reasons, partitioning of the one-mode projection of G and other standard bipartite graph clustering techniques (*e.g.,* Dhillion (2001) and Sun et al. (2009)) are rendered less effective. We instead employ a different tack and exploit the node similarities computed earlier. We transform G into a new weighted graph $G^P = (P, E^P, w^{sim})$ where $E^P = \{(u, v) \mid u, v \in P, sim(u, v) > \xi\}$, ξ is a predefined threshold, and w^{sim} is the edge weight function (*i.e.,* $w^{sim} = sim$). Thus, G^P is the similarity graph of document groups. $\xi = 0.5$ was used as the threshold for our analyses. To find clusters in G^P, we employ the Louvain algorithm, a heuristic method based on modularity optimization (Blondel et al., 2008). Modularity measures the fraction of edges falling within clusters as compared to the expected fraction if edges were distributed evenly in the graph (Newman, 2006). The algorithm initially assigns each node to its own cluster. At each iteration, in a local and greedy fashion, nodes are re-assigned to clusters with which they achieve the highest modularity.

[3] I is the index set of $L^{u,v}$.

4 Example Analysis: NSF Grants

As a realistic and informative case study, we utilize our model to characterize funding programs of the National Science Foundation (NSF). This corpus consists of 132,372 grant abstracts describing awards for basic research and other support funded by the NSF between the years 1990 and 2002 (Bache and Lichman, 2013).[4] Each award is associated with both a program element (*i.e.,* funding source) and a date. We define *document groups* in two ways: by program element and by calendar year. For comparison criteria, we used topics discovered with the MALLET implementation of LDA (McCallum, 2002) using $K = 400$ as the number of topics and 200 as the number of iterations. All other parameters were left as defaults. The NSF corpus possesses unique properties that lend themselves to experimental evaluation. For instance, program elements are not only associated with specific sets of research topics but are named based on the content of the program. This provides a measure of ground truth against which we can validate our model. We structure our analyses around specific questions, which now follow.

Which NSF programs are focused on specific areas and which are not? When defining *document groups* as program elements (*i.e.,* each NSF program is a node in P), node entropy can be used to answer this question. Table 1 shows examples of program elements most and least associated with specific topics, as measured by entropy. For example, the program *1311 Linguistics* (low entropy) is largely focused on a single *linguistics* topic (labeled by LDA with words such as "language," "languages," and "linguistic"). By contrast, the *Australia* program (high entropy) was designed to support US-Australia cooperative research across many fields, as correctly inferred by our model.

Low Entropy Program Elements	
Program	**Primary LDA Topic**
1311 Linguistics	language languages linguistic
4091 Network Infrastructure	network connection internet
High Entropy Program Elements	
Program	**Primary LDA Topic**
5912 Australia	(many topics & disciplines)
9130 Res. Improvements in Minority Instit.	(many topics & disciplines)

Table 1: **[Examples of High/Low Entropy Programs.]**

Which research areas are growing/emerging? When defining *document groups* as calendar years (instead of program elements), low entropy nodes in C are topics concentrated around certain years. Concentrations in

[4] Data for years 1989 and 2003 in this publicly available corpus were partially missing and omitted in some analyses.

later years indicate growth. The LDA-discovered topic *nanotechnology* is among the lowest entropy topics (*i.e.*, an outlier topic with respect to entropy). As shown in Figure 2, the number of *nanotechnology* grants drastically increased in proportion through 2002. This result is consistent with history, as the National Nanotechnology Initiative was proposed in the late 1990s to promote nanotechnology R&D.[5] One could also measure such trends using budget allocations by incorporating the award amounts into the edge weights of G.

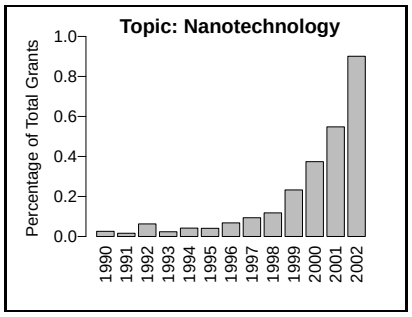

Figure 2: **[Uptrend in Nanotechnology.]** Our model correctly identifies the surge in nanotechnology R&D beginning in the late 1990s.

Given an NSF program, to which other programs is it most similar? As described in Section 3, when each node in P represents an NSF program, our model can easily identify the programs most similar to a given program. For instance, Table 2 shows the top three most similar programs to both the *Theoretical Physics* and *Ecology* programs. Results agree with intuition. For each NSF program, we identified the top n most similar programs ranked by our $sim(\cdot,\cdot)$ function, where $n \in \{3, 6, 9\}$. These programs were manually judged for relatedness, and the Mean Average Precision (MAP), a standard performance metric for ranking tasks in information retrieval, was computed. We were unsuccessful in evaluating alternative weighted similarity measures mentioned in Section 3 due to their aforementioned issues with scalability and the size of the NSF dataset. (For instance, the implementations of ASCOS (Antonellis et al., 2008) and SimRank (Jeh and Widom, 2002) that we considered are available here.[6]) Recall that our $sim(\cdot,\cdot)$ function is based on measuring the cosine similarity between two weight vectors, $\vec{a}$ and $\vec{b}$, generated from our bipartite graph model. As a baseline for comparison, we evaluated two additional similarity implementations using these weight vectors. The first measures the similarity between weight vectors using weighted Jaccard similarity, which is $\frac{\sum_k \min(a_k,b_k)}{\sum_k \max(a_k,b_k)}$ (denoted as *Wtd. Jaccard*). The second measure is implemented by taking the Spearman's rank correlation coefficient of $\vec{a}$ and $\vec{b}$ (denoted as *Rank*). Figure 3 shows the Mean Average Precision (MAP) for each method and each value of n. With the exception of the difference between *Cosine* and *Wtd. Jaccard* for MAP@3, all other performance differentials were statistically significant, based on a one-way ANOVA and post-hoc Tukey HSD at a 5% significance level. This, then, provides some validation for our choice.

[5] See *National Nanotechnology Initiative* at `http://en.wikipedia.org` for more information.

[6] See `networkx_addon` project at `http://github.com/hhchen1105/`.

1245 Theoretical Physics	**1182 Ecology**
1286 Elementary Particle Theory	1128 Ecological Studies
1287 Mathematical Physics	1196 Environmental Biology
1284 Atomic Theory	1195 Ecological Research

Table 2: **[Similarity Queries.]** Three most similar programs to the *Theoretical Physics* and *Ecology* programs.

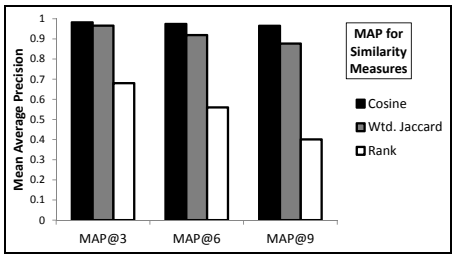

Figure 3: **[Mean Average Precision (MAP).]** Cosine similarity outperforms alternative approaches.

How do NSF programs join together to form larger program categories? As mentioned, by using the similarity graph G^P constructed from G, clusters of related NSF programs can be discovered. Figure 4, for instance, shows a discovered cluster of NSF programs all related to the field of neuroscience. Each NSF program (*i.e.*, node) is composed of many documents.

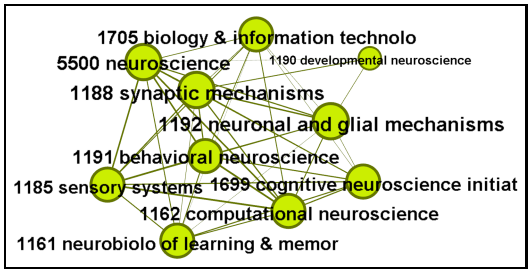

Figure 4: **[Neuroscience Programs.]** A discovered cluster of program elements all related to *neuroscience*.

Which pairs of grants are the most similar in the research they describe? Although the focus of this paper is on drawing comparisons among *groups* of documents, it is often necessary to draw comparisons among *individual* documents, as well. For instance, in the case of this NSF corpus, one may wish to identify pairs of grants from different programs describing

highly similar lines of research. One common approach to this is to exploit the low-dimensional representations of documents returned by LDA (Blei et al., 2003). Any given document $d_i \in D$ (where $i \in \{1 \dots N\}$) can be represented by a K-dimensional probability vector of topic proportions given by θ_{i*}, the i^{th} row of the document-topic matrix θ. The similarity between any two documents, then, can be measured using the distance between their corresponding probability vectors (*i.e.,* probability distributions). We quantify the similarity between probability vectors using the complement of Hellinger distance: $H_S(d_x, d_y) = 1 - \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{K}(\sqrt{\theta_{xi}} - \sqrt{\theta_{yi}})^2}$, where $x, y \in \{1 \dots N\}$. Unfortunately, identifying the set of *most* similar document pairs in this way can be computationally expensive, as the number of pairwise comparisons scales quadratically with the size of the corpus. For the moderately-sized NSF corpus, this amounts to well over 8 billion comparisons. To address this issue, our bipartite graph model can be exploited as a *blocking* heuristic using either the document groups or the comparison criteria. In the latter case, one can limit the pairwise comparisons to only those documents that reside in the same subset of D^C. For the former case, *node similarity* can be used. Instead of comparing each document with every other document, we can limit the comparisons to only those document groups of interest that are deemed similar by our model. As an illustrative example, out of the 665 different NSF programs covering these $132,372$ grant abstracts, the program *1271 Computational Mathematics* and the program *2865 Numeric, Symbolic, and Geometric Computation* are inferred as being highly similar by our model. Thus, we can limit the pairwise comparisons to only such *document groups* that are similar and likely to *contain* similar documents. In the case of these two programs, the following two grants are easily identified as being the most similar with a Hellinger similarity (H_S) score of 0.73 (only text snippets are shown due to space constraints):

> Grant #1
> *Program*: 1271 Computational Mathematics
>
> *Title*: Analyses of Structured **Computational Problems** and **Parallel** Iterative **Algorithms**.
>
> *Abstract*: The main objectives of the research planned is the analysis of **large scale** structured **computational problems** and of the convergence of **parallel** iterative methods for solving **linear systems** and applications of these techniques to the solution of large **sparse** and dense structured systems of **linear equations**

> Grant #2
> *Program*: 2865 Numeric, Symbolic, and Geometric Computation
>
> *Title*: **Sparse Matrix Algorithms** on **Distributed** Memory Multiprocessors.
>
> *Abstract*: The design, analysis, and implementation of **algorithms** for the solution of **sparse matrix** problems on **distributed** memory multiprocessors will be investigated. The development of these **parallel sparse matrix algorithms** should have an impact of challenging **large-scale computational problems** in several scientific, econometric, and engineering disciplines.

Some key terms in each grant are manually highlighted in bold. As can be seen, despite some differences in terminology, the two lines of research are related, as matrices (studied in Grant #2) are used to compactly represent and work with systems of linear equations (studied in Grant #1). That is, despite such differences in terminology (*e.g.,* "matrix" vs. "linear systems", "parallel" vs. "distributed"), document similarity can still be accurately inferred by taking the Hellinger similarity of the LDA-derived low-dimensional representations for the two documents. In this way, by exploiting the *group-level* similarities inferred by our model in combination with such document-level similarities, we can more effectively "zero in" on such highly related document pairs.

5 Conclusion

We have presented a bipartite graph model for drawing comparisons among large *groups* of documents. We showed how basic algorithms using the model can identify trends and anomalies among the document groups. As an example analysis, we demonstrated how our model can be used to better characterize and evaluate NSF research programs. For future work, we plan on employing alternative comparison criteria in our model such as those derived from named entity recognition and paraphrase detection.

References

[Antonellis et al.2008] Ioannis Antonellis, Hector G. Molina, and Chi C. Chang. 2008. Simrank++: Query Rewriting Through Link Analysis of the Click Graph. *Proc. VLDB Endow.*, 1(1):408–421, August.

[Bache and Lichman2013] K. Bache and M. Lichman. 2013. UCI machine learning repository.

[Blei et al.2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3(4-5):993–1022, March.

[Blondel et al.2008] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008+, July.

[Campr and Ježek2013] Michal Campr and Karel Ježek. 2013. Topic Models for Comparative Summarization. In Ivan Habernal and Václav Matoušek, editors, *Text, Speech, and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 568–574. Springer Berlin Heidelberg.

[Chen and Giles2013] Hung H. Chen and C. Lee Giles. 2013. ASCOS: An Asymmetric Network Structure COntext Similarity Measure. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 442–449, New York, NY, USA. ACM.

[Dhillion2001] Inderjit S. Dhillion. 2001. Co-clustering Documents and Words Using Bipartite Spectral GraphPartitioning. Technical report, Austin, TX, USA.

[Eagle et al.2010] Nathan Eagle, Michael Macy, and Rob Claxton. 2010. Network diversity and economic development. *Science*, 328(5981):1029–1031, May.

[Huang et al.2011] Xiaojiang Huang, Xiaojun Wan, and Jianguo Xiao. 2011. Comparative News Summarization Using Linear Programming. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 648–653, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Jeh and Widom2002] Glen Jeh and Jennifer Widom. 2002. SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 538–543, New York, NY, USA. ACM.

[McCallum2002] Andrew K. McCallum. 2002. MALLET: A Machine Learning for Language Toolkit.

[Newman2006] M. E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, June.

[Sarwar et al.2001] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA. ACM.

[Sun et al.2009] Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-based Clustering of Heterogeneous Information Networks with Star Network Schema. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 797–806, New York, NY, USA. ACM.

[Wan et al.2011] Xiaojun Wan, Houping Jia, Shanshan Huang, and Jianguo Xiao. 2011. Summarizing the Differences in Multilingual News. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 735–744, New York, NY, USA. ACM.

[Wang et al.2012] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2012. Comparative Document Summarization via Discriminative Sentence Selection. *ACM Trans. Knowl. Discov. Data*, 6(3), October.

[Zhai et al.2004] ChengXiang Zhai, Atulya Velivelli, and Bei Yu. 2004. A Cross-collection Mixture Model for Comparative Text Mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 743–748, New York, NY, USA. ACM.